

Pc-Check[®] UEFI

User Manual

Professional, computer diagnostic software,
fully UEFI-native pre-boot



**Assuring direct, legacy-free
UEFI hardware testing**

www.eurosoft-uk.com

Contents

Overview	4
Minimum System Requirements	4
A brief history of UEFI	4
How UEFI differs from legacy BIOS and DOS	5
The UEFI boot process	6
The Boot Manager	6
Secure Boot	6
Getting Started with Pc-Check UEFI Self-booting the diagnostics	8
The BIOS Boot Menu	8
Running diagnostics from the UEFI shell	8
Differences from Pc-Check EuroDOS	9
General	9
Group and Test Numbers	9
Test Parameters	9
Devices	9
Config.xml instead of cmdline.txt	10
Unicode	10
Navigation of the User Interface	11
The Main Menu	12
System Overview	12
Advanced Diagnostics	13
Run Testing Script	15
Advanced Scripting	15
View Results	18
Exit	18
Special Function Keys	18
Special Notes	19
Configuring Pc-Check UEFI	20
The GUI element	20
The FILES element	20
PciDeviceData	20
EventsFile	20
SysInfoFile	20

ResultFile	20
ResultsXSL.....	21
The FOLDERS element.....	21
Results	21
ReadOnlyMedia	22
OnExit	22
Help.....	22
DebugFile	22
File name and path macros.....	23
%STAMP%.....	23
%PROGDIR%	23
Notes	23
Troubleshooting.....	24

Overview

This document has been written to give information about the basic operation of Pc-Check UEFI. Pc-Check UEFI is run from USB drive, using it to boot PC systems. Pc-Check UEFI is a fast, accurate and easy-to-use diagnostics tool from Eurosoft that enables you to check the configuration and reliable operation of PCs.

Pc-Check UEFI is extremely useful to manufacturers and repairers of PCs, informing all Professionals of the architecture and operational reliability of PC hardware. It is assumed that readers have good technical knowledge and experience to gain full product usage. This manual is for standard Pc-Check UEFI features.

Minimum System Requirements

Compliant UEFI BIOS Revision 2.3.1 or later
64 bit
SMBIOS 2.7 or later
USB port
1GB RAM

Where possible, assuming it can start, Pc-Check UEFI will attempt to identify any unmet prerequisite requirements and add them to the event log.

A brief history of UEFI

UEFI stands for Unified Extensible Firmware Interface.

UEFI first came about as a bootstrap solution for Intel's new Itanium 64 bit processor in the late 1990's. Pioneered by Intel, Microsoft, HP and others, it was called IBI - the Intel Boot Initiative.

When the PC market instead adopted 64 bit extensions to the traditional x86 architecture, legacy BIOS persisted, but it was clear, with more and more powerful systems arriving, that this traditional BIOS approach was struggling to keep up, with a plethora of creative 'bolt-on' extensions and obscure standards.

Intel had continued to develop the IBI, by now named EFI, and added a conversion layer for x86 systems called the CSM (Compatibility Support Module). CSM gave x86 operating systems and software a legacy BIOS interface such that they did not need to be aware of EFI at that time. This, typically 32 bit EFI version, was in many PC systems, especially servers, during the 2000's, but because of the inclusion of CSM, this was not immediately obvious to users.

Intel contributed the EFI specification to the UEFI forum in 2005 to continue development, with the organisations first specification update released in 2007.

UEFI finally came into its own for the PC, once both 64 bit processors and 64 bit operating systems became the norm, and storage media capacity had exploded past the limitations of

legacy BIOS, especially MBR partition table encoding.

How UEFI differs from legacy BIOS and DOS

Executable files have the extension .efi instead of .exe

Batch (shell scripts) have the extension .nsh instead of .bat

Drives are not named by letters, such as A:, C:, but rather with short device names with a numeric component, such as FS0:, FS1:, where FS means 'File System'. Use of drive names is for the most part avoided in Pc-Check UEFI because the path assigned to the boot device can vary from one system under test to another, including the boot device from which the diagnostic is running.

The UEFI boot process

The Boot Manager

Legacy BIOS has a simple process for booting media - it simply loads machine executable code directly from the boot sector (first sector or Master Boot Record) of the boot disk and then calls it. In turn that code will inspect the partition table and upon determining the active partition, it will also simply load code from the boot sector (this time the first sector of the partition) and call it directly. For the sake of this discussion, the job of the BIOS is a simple one, and the necessary complexity for finding and loading boot files from file systems is all in the boot code. This approach gives almost total freedom to the boot code to behave however it might wish, and indeed entirely custom boot structures, proprietary partitioning schemes are possible, however to work seamlessly with other operating systems and disk management tools, boot code must observe and preserve certain data structures such as the location and format of the MBR partition table, which ultimately had limitations.

Since PC systems now have very much more firmware storage available, both for code and for system settings, UEFI BIOS instead have a built-in boot manager. The code includes a FAT file system driver, which is why operating systems on UEFI systems include an EFI system partition formatted with FAT to store the OS initialisation code. When creating media to boot on a UEFI system, the inherent ability to understand the FAT file system means that there are no special custom boot sectors, weird tools, or procedures, but just to provide the correct files in the correct folders that the BIOS will search for. (On a PC that is the file `EFI\BOOT\BOOTX64.EFI`). If you've had a long relationship with PC systems, this can seem as if some steps are missing.

Secure Boot

Digital signatures are a mathematical scheme by which to demonstrate the authenticity of files, especially executable code. A digital signature key identifies the organisation that created the file and so it is possible to make decisions as to whether that code should be allowed to execute. Any attempt to modify the executable code affects the validity of the digital signature and so malware infection or other tampering can be detected.

Secure Boot uses digital signatures to ensure that the code used to boot an operating system has not been modified or replaced. A list of trusted keys is held by the BIOS such that even if a file has a valid digital signature (has not been modified), it will not execute if the organisation that created it is not on the approved list. Keys can be added to this list, but to do so on every target PC to run diagnostics is cumbersome and the PC owner may not approve of introducing new keys to the system. It is possible to disable secure boot in most BIOS, and this is necessary if you wish to use the UEFI shell, however, once again this takes time, differs in method, and you must remember to re-enable secure boot after testing. Once again, the owner of the PC may not permit you to do this.

Thankfully there is a solution. As one might expect, Microsoft have keys in the default trusted database of UEFI PC systems. They have also set up a system for the submission of code to be countersigned by their digital signature such that it can be booted on any PC using the Microsoft default trusted key. Naturally there is a set of strict requirements that must be met before they will sign any such file submitted from another organisation. Eurosoft have

met these requirements with our UEFI diagnostic boot loader and so can offer the same 'walk up and test' self-boot convenience that was available to users of Pc-Check EuroDOS. There is no requirement to disable Secure Boot to run Pc-Check UEFI.

Getting Started with Pc-Check UEFI Self-booting the diagnostics

The BIOS Boot Menu

The simplest, most convenient, and recommended way to start Pc-Check UEFI is through use of the BIOS Boot Menu. Typically the Boot Menu is accessed by pressing a function key as the system starts at power on, commonly F12 or F11 is used. The correct key to use is frequently displayed as the system boots and at the correct time to press it, although if you know the key, it can help to start pressing it pre-emptively.

On many PCs you can also access a one-time 'Boot Override' menu from within the BIOS Setup screen. As the Microsoft Windows Settings and the Microsoft Windows Recovery Tools offer an alternative method to enter the BIOS Setup via their menus, this is especially handy if a system is configured for a 'fast boot' mode that makes accessing the boot menu difficult.

Having successfully invoked the BIOS Boot Menu, you will be presented with a list of the detected devices that can potentially be of use for system boot. If support for legacy mode is enabled in the BIOS, some of these devices may appear more than once. In this case, you should pay attention only to the entries that have a prefix of UEFI: the entries without this prefix shall boot the same device in legacy mode.

A Note on Dual Booting Media

Pc-Check UEFI may have been supplied to you on a dual boot device also containing Pc-Check EuroDOS. If you see the EuroDOS sign-on messages, you are booting in legacy mode. If you would prefer to test the system using the UEFI diagnostics, you should restart and be sure to observe the UEFI: prefix in the BIOS boot menu and if necessary ensure that a legacy only configuration has not been selected in BIOS Setup.

Running diagnostics from the UEFI shell

The shell can perform unrestricted and potentially destructive actions, some forms of the built in commands can even 'brick' a PC. It can run batch operations at start-up with a script named startup.nsh, much like DOS would run autoexec.bat. For these reasons, the UEFI command line shell should never be available in digitally signed form, and so if you wish to run diagnostics using the UEFI shell you must first disable the Secure Boot feature of the target system and keep media safe.

Pc-Check can be started from the UEFI shell and from within a shell script file. If the file structure of the diagnostics is reorganised, be sure to make matching adjustments to the config.xml file so that it knows where to find required files and where to place results files. If you want to stay in the shell or script after the diagnostics complete, be sure to change the OnExit element to "Exit".

In a shell script you have the flexibility to add other tools to the test sequence, move files, enable remote control and so on. If you need Pc-Check UEFI to run more than once with different settings and/or scripts you can keep multiple versions of the config.xml by different names and copy them into place before each execution.

Differences from Pc-Check EuroDOS

General

Aside from the clear differences in presentation, the fundamental internal operation of Pc-Check has evolved and has been aligned to Eurosoft's Windows diagnostics. This allows a best tool fit for each testing requirement, while sharing common scripting and results formatting. It also permits for automated dual environment testing with Pc-Check Windows able to invoke the diagnostics of Pc-Check UEFI (for more information please see the appropriate Pc-Check Windows manual or contact Eurosoft).

Group and Test Numbers

In Pc-Check EuroDOS, the notion of Groups (Hard Disk, Processor etc) and Tests (Linear Read, Core etc) exist, but were identified in the XML scripts and results by four character mnemonics or acronyms. Groups and tests are now identified by simple numerical IDs. Just like Pc-Check EuroDOS, once a value has been used it is typically not used again by Pc-Check UEFI for a different test, even if a test or group is deprecated.

Test Parameters

In Pc-Check EuroDOS, many tests have command line options to control some aspect of testing, for example to set the maximum duration for all hard disk media tests. In Pc-Check UEFI this is handled instead by test parameters held in the script itself. Test parameters accompany each individual test such that if, in the given example, more than one media test is selected, each test could run with a different duration setting. This also removes the need for the test pre-sets, where the same test might exist in multiple forms of Quick, Standard, Full and Custom.

Devices

In order to align with the way diagnostics work in Eurosoft's Windows products, Pc-Check UEFI assigns the testable units, the devices, differently in some cases. For some groups, where a sub-system might have been treated as a whole, this might now be available as several individually addressable items, while in other groups, a number of devices may have been combined into a single testable unit.

A good example of such a change is the Processor group. In Pc-Check EuroDOS, the Processor group assigns devices to logical processors, one for each core or hardware thread. In Pc-Check UEFI, the diagnostics instead assign devices to each processor package, such that there is normally only one device except for a multi-socket system such as a high end workstation or a server. With core and thread counts rising rapidly, this provides simpler, more compact results output and simpler scripting.

This does not however mean that the diagnostics are in any way less capable or have reduced coverage. In this Processor group example, it simply means that when CPU tests are

executed, the testing shall be applied automatically to all the cores and threads of the selected CPU package to produce a single package result.

Pc-Check EuroDOS has a special device number, -1, used in scripts that acts as a 'wild card' or 'test all'. It instructs the group to apply that test to all the detected devices. Similarly Pc-Check UEFI has such a special device number, but in this case it is 0.

Config.xml instead of cmdline.txt

Since, by design, EuroDOS does not provide a command prompt, EuroDOS instead passes command line options to Pc-Check through the use of a ASCII text file called cmdline.txt. If Pc-Check is used in another DOS environment with a command prompt, the options would not come from cmdline.txt and need to be provided as part of the command in the normal manner. Should there be a lot of options, this might mean that a .bat file is used to avoid retyping the options repeatedly.

In Pc-Check UEFI, global options are held in an XML file called config.xml. There are less global options needed because the use of Test Parameters replaces very many. Pc-Check UEFI always reads the config.xml file for settings, even when it is started from a shell prompt.

Unicode

All Pc-Check UEFI XML config, script and results files are encoded as Unicode (UTF-16) with BOM (Byte Order Mark). Pc-Check EuroDOS files are 8 bit ASCII.

Most popular text editing tools, including Windows NotePad, now automatically and seamlessly detect and edit Unicode files. Should you experience problems with Pc-Check UEFI - especially start-up issues after modifying config.xml - please make sure that your text editor of choice is properly configured to preserve or create the BOM.

Navigation of the User Interface

As a native UEFI application, keyboard, mouse and even touch input are available. At the time of writing, some laptops still don't provide support for the built-in mouse (pad), but an external USB mouse is near universally supported, even in these cases. The mouse centre 'scroll wheel' is not supported by UEFI BIOS.

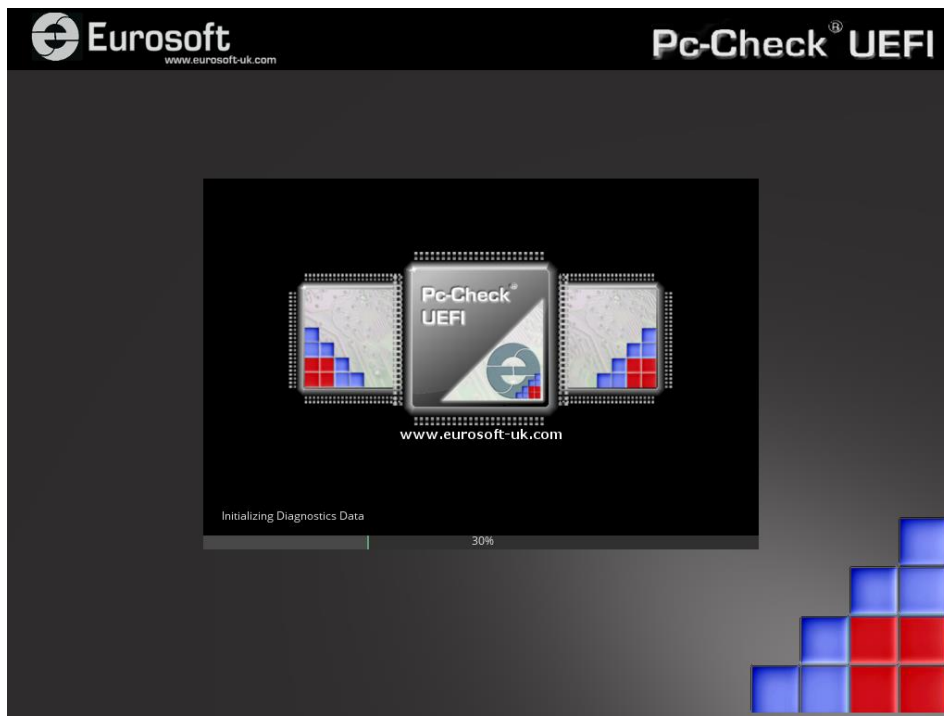


Users of the Pc-Check EuroDOS will find that the initial menus can be navigated using the keyboard in a very similar manner. Cursor keys move between menu options, while Enter selects the highlighted option and Escape backs out to the previous menu. The Escape key will highlight the Exit option on the Main Menu. On screens where the graphical user interface necessarily becomes more complex, keyboard input takes inspiration from other Windowing systems, additionally using the Tab key to move between controls and space bar to make selections.

Touch devices are supported. Touch events are handled as if the mouse moved directly to that point and the left button was clicked. There are no items in Pc-Check UEFI that require the right mouse button.

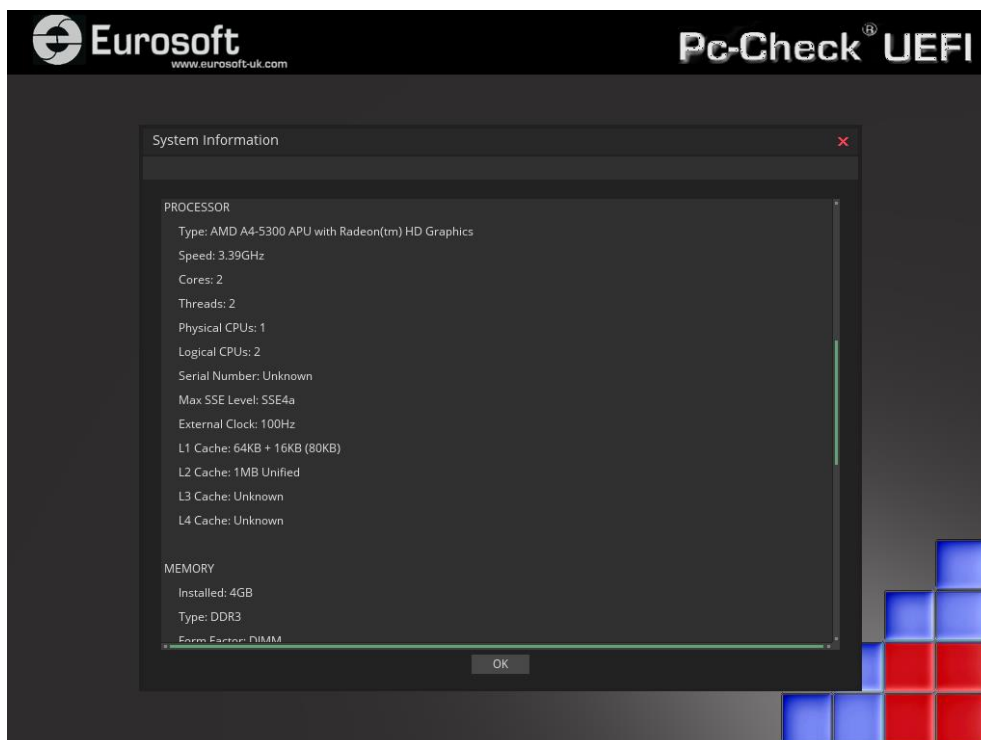
Clicking the X icon in the top right hand corner of a Window will cancel that action and take you back.

The Main Menu



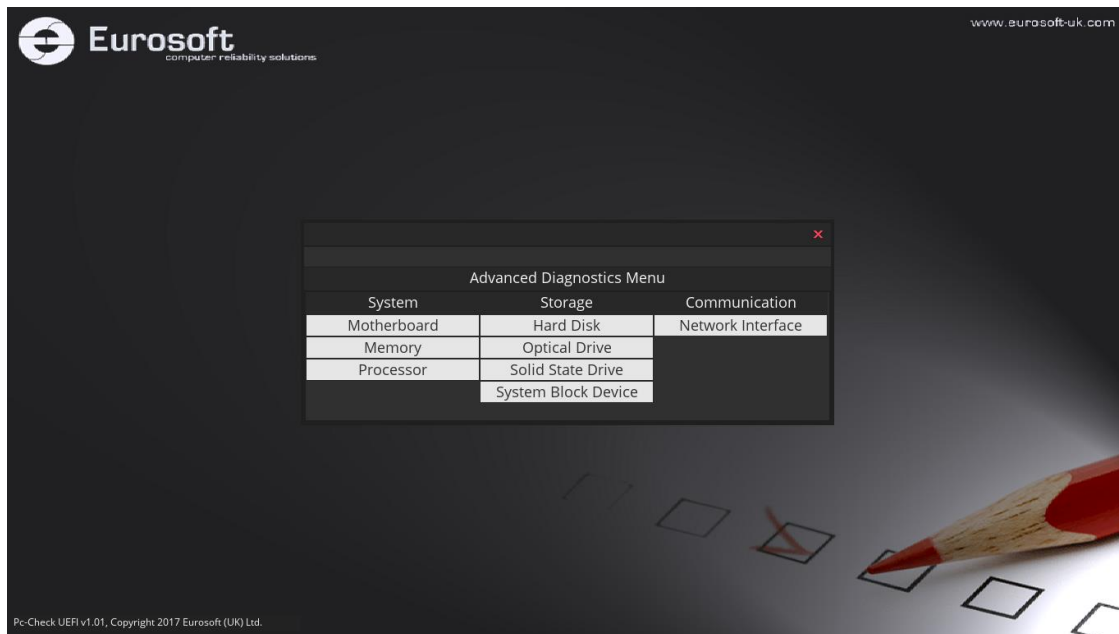
System Overview

Characterises the PC system by identifying and listing the main components and capabilities to the display. Detailed system information is also captured to the Results folder in a file named SysInfo.xml.

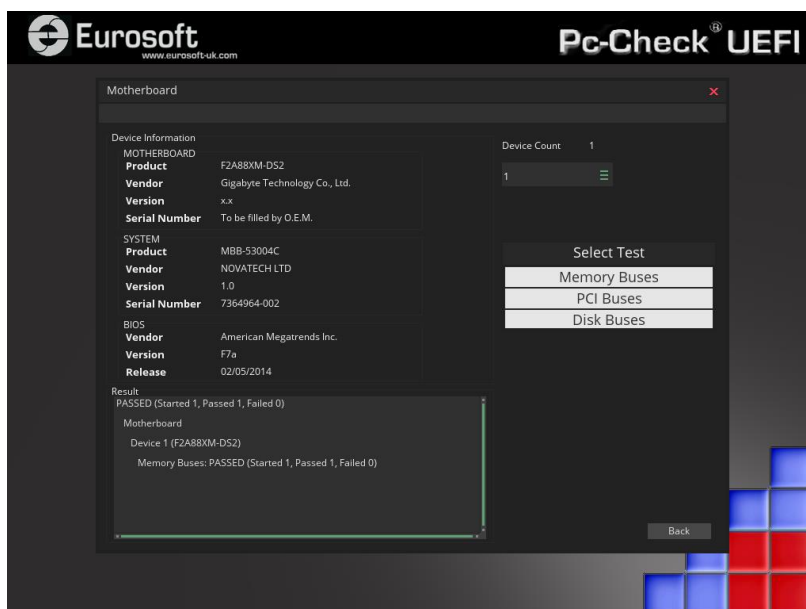


Advanced Diagnostics

This invokes the Advanced Diagnostics Menu. Use the Advanced Diagnostics menu to run tests in a free-form, exploratory manner. Each of the groups has an entry in the menu, which is arranged by category. Groups that have not detected testable devices will be unavailable and 'greyed out'. In some cases, such as for the Optical group, devices require media to be pre-loaded at initialisation to be considered testable. On selection of a Group, you will be taken to the Group window.



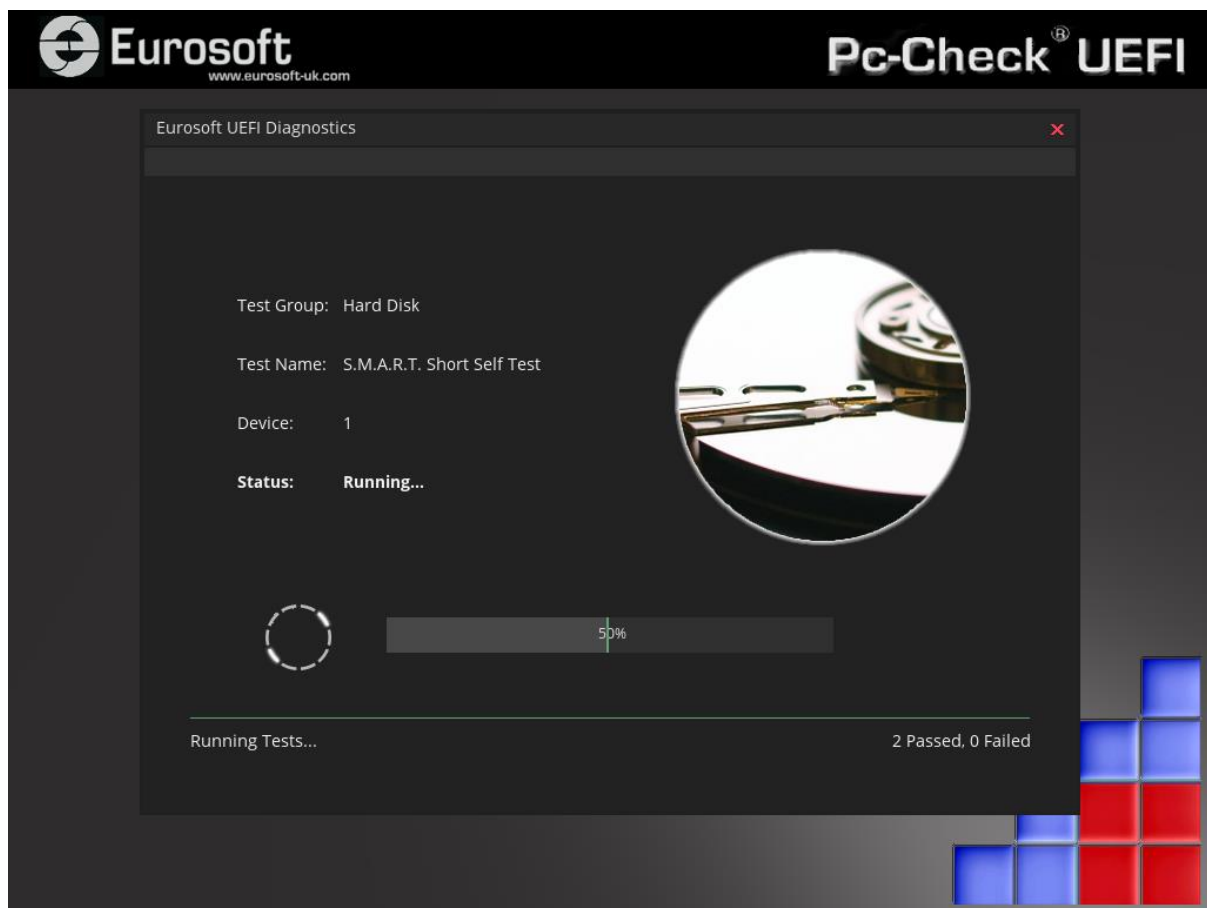
On entry to the Group window, if there are multiple testable devices available, the focus for keyboard input will be directed to the device number selection on the top right. Use the up and down cursor keys to select the device number to test. Information about the selected device is shown on the left side and will be updated as the device selection changes. Once the correct device is showing, press the Tab key to move focus to the Select Test menu.



After device selection (or if only one testable device is available), the focus for keyboard input will move to the Select Test menu on the bottom right. If you are using keyboard, use the cursor keys and press Enter to select a test. Choosing a test will take you to the first of two Test Details windows.

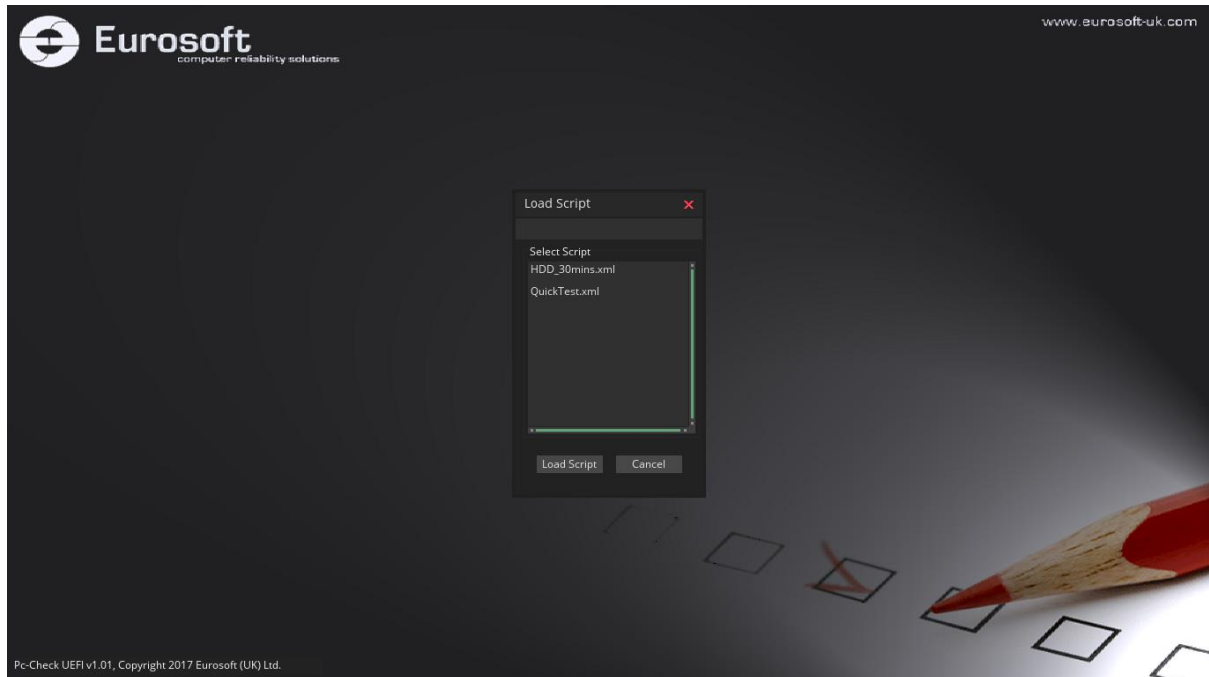
In the first Test Details Window, Test Parameters can be set - for example, you can set the duration of a storage media test. Select the Test Parameters from the list on the left hand side and modify the values shown on the right. If you just wish to continue immediately with the default settings, the keyboard focus is directed initially to the Ok button, and so you may just press Enter. You will be taken to the second Test Details Window.

In the second Test Details Windows you can set a Lap Count and indicate if testing should stop (or else continue) after a test failure. By default the test will be run once only. Again, since keyboard focus is directed initially to the Ok button, another press of the Enter key will take you directly to testing.



Run Testing Script

Lists the available test scripts and immediately executes the selected script. By default scripts are held in a folder named 'Scripts' found in the root of the Program Device. The location of this folder can be set by changing the Scripts element in config.xml.

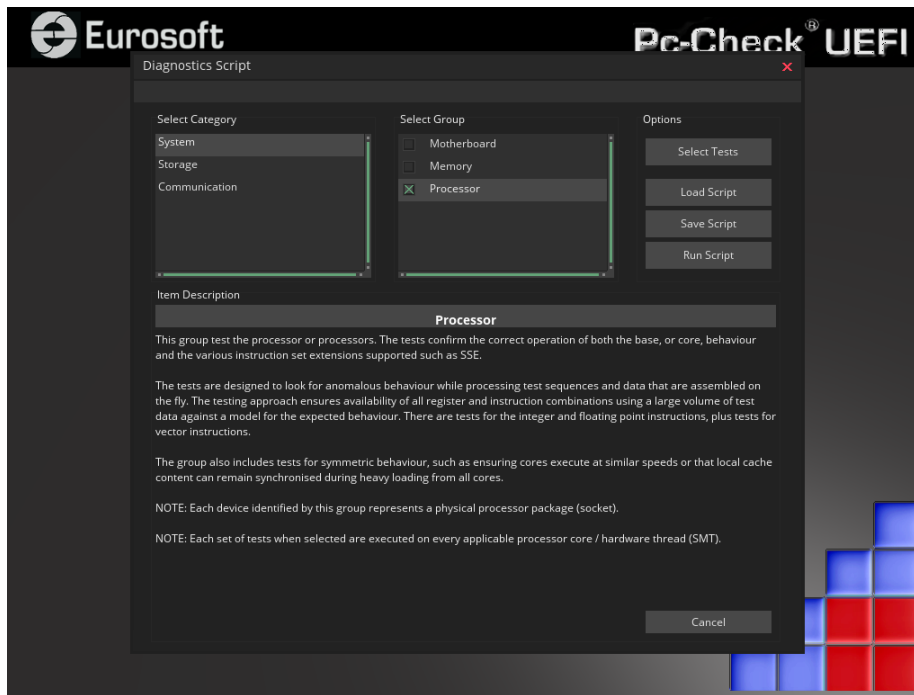


Advanced Scripting

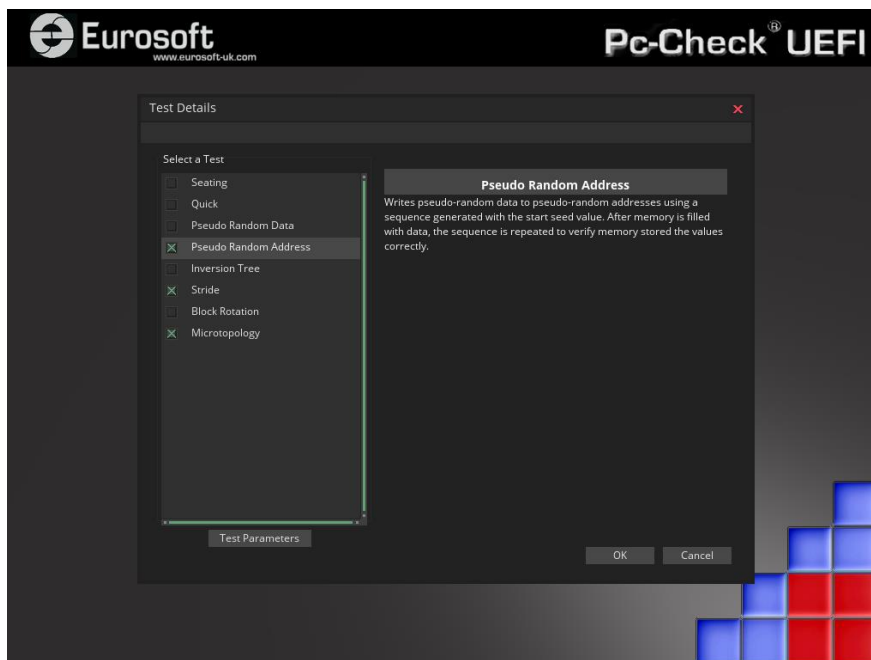
For the creation, execution and saving of test scripts. Script creation is presented a simple hierarchy of list selection, with the same basic modes of operation at each level.

When selected for the first time during a session, test help information including group, test and parameter descriptions are first loaded. This will not occur if the help settings are not at the default, intermediate level. You will then be presented with the Diagnostic Script window.

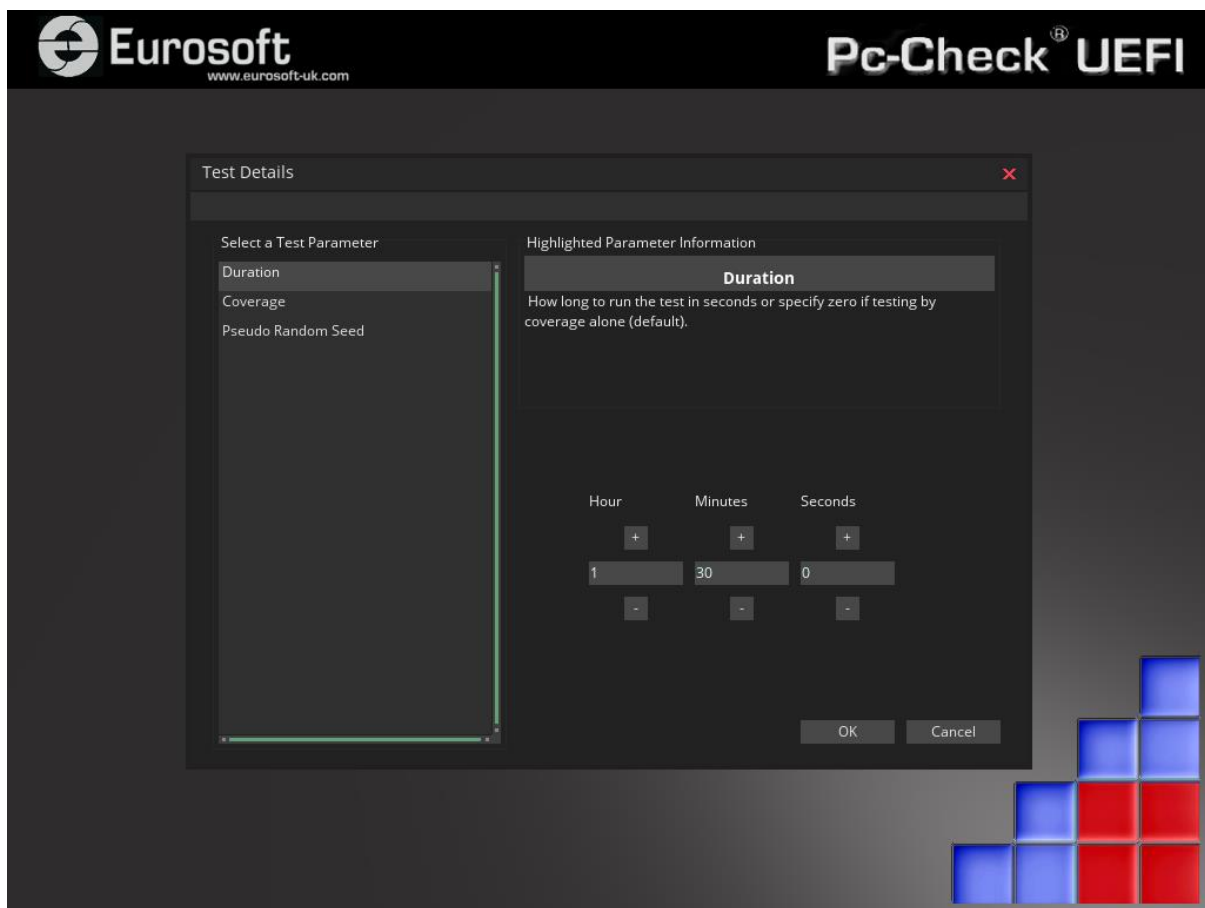
Work from the left to the right of the Diagnostic Script window. Start by selecting a Category from the first list, such as System. A list of Groups for this category will then be shown in the next list, toward the centre. Highlighting a Group from this list, such as Motherboard, will give help information for the Group, shown beneath. To select a Group for testing, activate the check box alongside it. You should now choose which tests shall be run for the selected Group - use the 'Select Tests' button from the top of the column of buttons to the right.



Test selection operates in the same manner as Group selection. Tests can be highlighted in a list on the left hand side of the Window to see a description of that test, shown to the right. To include the test in your script, select the check box alongside it. Beneath the list of tests is a button marked Test Parameters. With the test both highlighted and selected via the checkbox, use the Test Parameters button to view and modify the test behaviour, such as setting a test duration. If there are no Parameters for the test, a pop-up message will let you know. If the button appears to do nothing, be sure that a test is both highlighted and selected (its box is checked).



Once more, available Parameters are listed in a column. Selecting a Parameter from the list provides a description to the top right and the opportunity to view and modify the current value to the lower right.



When you have made all the test selections for your script, use the OK buttons to return from your current position until you reach the first, Group selection window. In the column of buttons to the right hand side of this window you may then choose to either save or run your script.

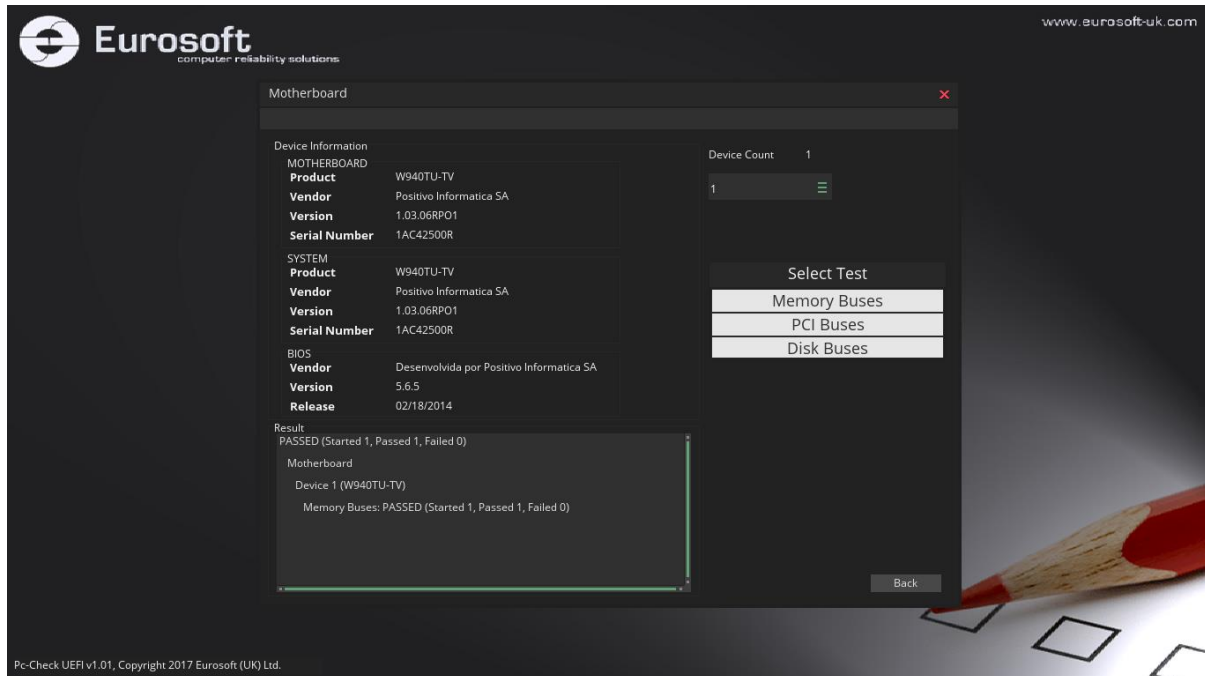
When you run or save a script, you will be prompted for the lap count and whether testing should continue after a test fails. This information is included in the script file when you save it. Should you run the script using the Main Menu option Run Testing Script, or automatically start the script from config.xml, these settings will be used.

Existing scripts can be loaded for modification and then can be run or saved. Overwrite the existing script name to permanently modify the script or saved as a new script with a new name. Scripts are loaded and saved from the Scripts folder. The location of this folder can be configured in config.xml.

Scripts test all detected devices for a Group. Scripts are written with the 'wild card' or 'test all' device value of 0. Although the need rarely arises, experienced users can manually modify the script to limit testing to specific device numbers or to vary parameters between devices, perhaps to best suit a fixed configuration of a specific product.

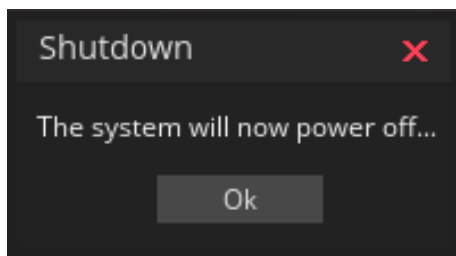
View Results

Lists the results for all the tests run during the session. Failed items are highlighted and prefixed with an exclamation point ! at the start of the line. The display will also show a 'traffic light' banner for the system status as a whole, provided there is at least one test with a pass or fail result. The results are also displayed on Exit from the program.



Exit

Displays the session results and then exits the program. Exit behaviour depends on how the program is configured. The system will either be shut down, rebooted or exit to prompt (shell operation).



Special Function Keys

F4 - Closes the currently active window, as if the X icon has been clicked. This works even in situations where the Escape key might not.

F5 - Refresh the display. In the unlikely event the user interface has not redrawn properly, this key requests that everything be redrawn.

F9 - Capture the screen to a bitmap file. The file will be given a date and time stamp such that you can capture multiple screens. Files are written alongside the main executable (.efi) file, usually this will be in the EFI\BOOT folder. You might do this to send an image to technical support for example.

Special Notes

Although Eurosoft has gone to extraordinary lengths to provide an appealing and simple graphical user interface, customers are reminded that they are still in the pre-boot environment, not a fully featured operating system. This provides significant advantages to the diagnostic testing, but as UEFI is not a multi-threaded environment, some of the responsiveness and luxuries that we ordinarily expect are not always exhibited. At times, the mouse pointer may be temporarily removed from the display to avoid confusion where the system might otherwise seem unresponsive. This is usually preceded by a short animation to show that the system is busy. Please allow time for the system to pick up keyboard requests for actions such as aborting tests or making screen captures.

Configuring Pc-Check UEFI

Pc-Check UEFI is configured using the XML file config.xml. This file can be found alongside the main program executable (.efi) file, usually in the folder EFI\BOOT.

The file is made up of a few different main sections or 'elements' in XML speak. Within each section there are further XML elements that configure various program settings or behaviours. Each of these elements has one or more XML attributes (not to be confused with test attributes) which you may configure for the desired outcome. Example of this would be file="Results.xml" or action="ShutDown".

The config.xml includes embedded comments (English language) that give helpful brief reminders of the purpose of each element and the available values for the attributes.

The GUI element

This will be preconfigured for you and should not be changed unless directed by Technical Support.

The FILES element

This configures the names of files needed or created by the program. (Also see the FOLDERS element.)

PciDeviceData

This is the location of the PCI device database file.

EventsFile

This is an XML output file from the diagnostics which records significant events, useful general information and errors from the program (not the same as test failures). This file will be created in the folder specified by the FOLDER element named Results.

SysInfoFile

This is an XML output file from the diagnostics that captures a snapshot of the system configuration as detected by the test groups. These are referred to as the device attributes. This file will be created in the folder specified by the FOLDER element named Results.

ResultFile

This is an XML output file from the diagnostics that records the results of the diagnostic testing. As each test is executed, the results file logs the start and end of each test along with the outcome. This file will be created in the folder specified by the FOLDER element named Results.

ResultsXSL

This file path gets inserted into the results file XML header as a (stylesheet) reference. This affects the filtering and formatting of the results when viewed in a browser. An example XSL file is provided to format the results as a HTML table. If the ResultsXSL element is omitted or set to an empty value, "", an XSL reference will not be added to the results XML file. Note that the Google Chrome browser will not display results formatted with an XSL file held on a local file system unless started with option --allow-file-access-from-files, it will instead simply show an empty tab. As this file path is used by a browser while in a full operating system such as Windows, it should be configured appropriate to that environment, and not a UEFI path.

ScriptFile

This is an XML input file for the diagnostics. It describes a series of tests actions, test parameters and global actions to be taken. When specified, this script file will automatically be started as soon as Pc-Check UEFI has loaded. After testing, the program will stop at the results display page and from there will exit. If the ScriptFile is not present or is an empty string "", the Main Menu will be shown. The script file should be stored in the folder specified by the FOLDER element named Scripts.

The FOLDERS element

This configures the names of folders needed or created by the program.

Results

This is where Pc-Check UEFI will store the file specified by the FILES elements EventsFile, ResultFile and SysInfoFile. By default these files are named Events.xml, Results.xml and SysInfo.xml and so existing files in this folder will be overwritten by files of a new test session. To prevent overwrite you can add the macro %STAMP% to give these files unique session names, for example;

```
<EventsFile file="%STAMP%_Events.xml">Events File</EventsFile>
<SysInfoFile file="%STAMP%_SysInfo.xml">System Information</SysInfoFile>
<ResultFile file="%STAMP%_Results.xml">Test Results</ResultFile>
```

Scripts

This is the folder where Pc-Check UEFI test scripts are stored. If a script is listed, loaded or saved, it will be from this folder.

Images

This directs Pc-Check UEFI to data used by the Graphical User Interface. It is not normally necessary to change this setting unless you were restructuring the file system for a very specific requirement, perhaps when scripting in a shell.

The OPTIONS element

This configures various behaviours of the program.

OnBreak

This element specifies the key that interrupts testing and how that shall be handled. Available actions include ending the script, skipping the current test or it can prompt for the user to choose what action to take.

ReadOnlyMedia

When set to true, this element tells Pc-Check UEFI not to create any output files. Set the value to true if you have a requirement to use protected media and so to prevent error messages.

OnExit

This element tells Pc-Check UEFI what action to take when the program exits. The system can be shut down, rebooted or the program can simply exit. Choose the exit option if you are running the diagnostics in a shell environment and want to return to the prompt or to an executing shell script. It is best to avoid the exit option when running the software self-booted as although the UEFI specification dictates that the boot loader should move to the next boot device and so start the installed operating system without POSTing, individual BIOS behaviour is unpredictable, use the reboot option to achieve the same option with BIOS POST step.

Help

This element tells Pc-Check UEFI if and when to load help information. When shown, help information gives descriptions of tests and their parameters. By default the level is Intermediate, meaning that the help information is not loaded until the Advanced Scripting option is used to create, run and save test scripts. Alternative settings are Beginner, which requests that the help information be loaded before the Main Menu, and Expert, which stops help information being loaded at all, allowing for faster operation. When enabled, help information only loads once per session.

DebugFile

At start-up, should Pc-Check UEFI fail to reach a stable state, that is should it fail to display the Main Menu or to begin executing a test script, on the next start-up it will capture a debug log file. If the problem has cleared or Pc-Check UEFI has been booted on a new system on which it reaches the stable state, the debug log file is erased. This facility will be unavailable if ReadOnlyMedia is specified or the media cannot be written. Typically changes to these options would be made on the advice and with the assistance of Technical Support.

File name and path macros

%STAMP%

Using %STAMP% in a file path or file name will see the macro replaced with a time-stamp of the start of the diagnostics session. This can be used to automatically make a separate copy of results from each diagnostic run that do not overwrite the previous results. The time-stamp will take the form D20170431T235959, that is 'D', year(4digits), month(2digits), day(2digits), 'T', hour(2digits), minute(2digits), second(2digits) with no spaces or other separators.

%PROGDIR%

Using %PROGDIR% in a file path will see the macro replaced with the absolute path to the main program executable (.efi) file on the program device. For a typical self boot program device this will be \EFI\BOOT\

Notes

Config.xml is a UniCode (UTF-16) XML file with a BOM (Byte Order Mark). Please make sure that your text editor is properly configured to preserve or create the BOM.

Windows Notepad works just fine for editing the config.xml file. If you use a more sophisticated replacement for your text edits, please ensure that it can handle Unicode files and is correctly configured to have the setting 'Write a byte order mark at the start of Unicode files' (or similarly named option) enabled.

All paths in Pc-Check UEFI are to the program device. Where a path starts with a backslash (\), it will be from the root of that device - for example \EFI\BOOT. A file system designator such as FS0: should not be included as this can vary between different systems under test.

Troubleshooting

Not booting to Pc-Check UEFI

- Select the correct USB entry from the boot menu.
 - UEFI boot devices typically have 'UEFI:' prefix in the device list.
- Is it a UEFI PC?
 - The UEFI diagnostics requires a PC with a UEFI BIOS.
- Is UEFI disabled in the BIOS. Check settings for 'Legacy Only' or similar setting.
- Microsoft Surface Products
 - Your system has the 'Microsoft Only' secure boot setting. Please change the setting to 'Microsoft and approved 3rd parties'.
- Not getting to the Main Menu
 - Going straight to a test script?
 - Check that the config.xml file is not set to autorun a script.
 - Check for licensing errors.
 - Try booting a second time and look for an error log on your boot device.
 - The error log will be called debug.txt.

It is created when Pc-Check UEFI detects that it failed to reach the main menu on the previous boot attempt. If that next boot attempt is successful, then the debug.txt file is removed. Technical support might request the debug file as part of any investigation.

- License errors
 - Is the license file present?
 - Is your Eurosoft USB program device inserted?
- Font errors and display issues
 - Make sure that your config.xml file is present.
 - Found alongside the main executable (.efi), typically in folder EFI\BOOT
 - Make sure that your config.xml file is valid.
 - Config.xml is a UniCode (UTF-16) file with a BOM (Byte Order Mark). Please make sure that your text editor is properly configured to preserve or create the BOM.
- Testing issues
 - Optical group is not available.
 - The Optical device group requires that suitable media be pre-loaded when the diagnostics initialise. Insert a data disc with as near full capacity for it type as possible and restart the diagnostics. Choose good quality (ideally pressed) media that had been carefully handled since you wish to test the drive hardware and do not want results to be affected by poor media quality.
- Network group is not available.
 - BIOS Network support is commonly disabled in the default shipping state of PCs. Use the BIOS Setup to enable Network support.
- Mouse and touch issues
 - The laptop built-in mousepad does not work in Pc-Check UEFI.
 - Plug in an external USB mouse. Surprisingly, some systems don't provide support for their built-in mousepad device, but do provide support for an external USB mouse. We suggest that technicians should carry a small USB mouse.

- The centre scroll wheel of the mouse doesn't work.
 - This is normal. This button is unsupported in the pre-boot UEFI environment.
- Touch screen doesn't work.
 - The BIOS has not implemented the touch screen in pre-boot. Pc-Check has support for touch screen 'absolute' devices where the BIOS provides it, but support is not universal amongst touch capable devices.
- Viewing results
 - Empty page when viewing results in Google Chrome
 - The Google Chrome browser implements a security feature that prevents the use of the local file system for XSL stylesheets. The specific threat is considered somewhat tenuous by many and the lack of any error or warning in normal operating mode means it is not obvious why the content has not displayed. Choices are to open Chrome with the command line flag `--allow-file-access-from-files`, open the results file in another browser, such as Internet Explorer, or to remove the stylesheet reference from the results file. To permanently disable XSL formatting, edit the ResultsXSL element in config.xml to an empty string, "".